(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
1 June 2006 (01.06.2006)

PCT

(10) International Publication Number
WO 2006/058173 A2

(51) International Patent Classification:
H04J 3/12 (2006.01)

(21) International Application Number:
PCT/US2005/042643

(22) International Filing Date:
23 November 2005 (23.11.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/630,853 24 November 2004 (24.11.2004) US
60/632,825 2 December 2004 (02.12.2004) US

(71) Applicant *(for all designated States except US)*: **QUALCOMM INCORPORATED** [US/US]; 5775 Morehouse Drive, San Diego, CA 92121-1714 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **KATIBIAN, Behnam** [US/US]; 28 Cedar Ridge, Irvine, California 92603 (US). **WILEY, George, A.** [US/US]; 5740 Brittany Forest Lane, San Diego, California 92130 (US).

(74) Agents: **KESSLER, Edward** et al.; STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C., 1100 New York Avenue, N.w., Washington, DC 20005 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

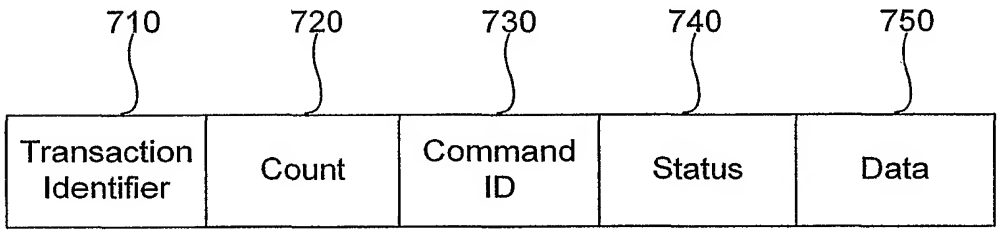
(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:
— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: DIGITAL DATA INTERFACE DEVICE MESSAGE FORMAT



(57) Abstract: The present invention provides a digital data interface device message format that describes command and response messages to be exchanged between a digital device having a system controller and a digital data interface device. The digital data interface device includes a message interpreter, content module and a control module. The digital data interface device may include an MDDI link controller. The digital data interface device can be used by a cellular telephone to control a peripheral device, such as a camera, bar code reader, image scanner, audio device or other sensor. The digital data interface device message format includes a transaction identification field, a count field, a command identification field and a status field. Optionally, the message format can include a data field. When an MDDI link is used, a digital data interface device message can be included in an MDDI register access packet.

WO 2006/058173 A2

DIGITAL DATA INTERFACE DEVICE MESSAGE FORMAT

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] The present invention relates generally to data communications. More particularly, the invention relates to a digital data interface device message format.

Background

[0002] Computers, mobile telephones, mobile telephone cameras and video capture devices, personal data assistants, electronic game related products and various video technologies (e.g., DVD's and high definition VCRs) have advanced significantly over the last few years to provide for capture and presentation of increasingly higher resolution still, video, video-on-demand, and graphics images. Combining such visual images with high quality audio data, such as CD type sound reproduction, DVDs, and other devices having associated audio signal outputs, creates a more realistic, content rich, or true multimedia experience for an end user. In addition, highly mobile, high quality sound systems and music transport mechanisms, such as MP3 players, have been developed for audio only presentations to users.

[0003] The explosion of high quality data presentation drove the need to establish specialized interfaces that could transfer data at high data rates, such that data quality was not degraded or impaired. One such interface is a Mobile Display Digital Interface (MDDI), used, for example, to exchange high speed data between the lower and upper clamshells of a cellular telephone that has a camera. MDDI is a cost-effective, low power consumption, transfer mechanism that enables very-high-speed data transfer over a short-range communication link between a host and a client. MDDI requires a minimum of just four wires plus power for bi-directional data transfer that delivers a maximum bandwidth of up to 3.2 Gbits per second.

[0004] In one application, MDDI increases reliability and decreases power consumption in clamshell cellular telephones by significantly reducing the number of wires that run across a handset's hinge to interconnect the digital baseband controller with an LCD display and/or a camera. This reduction of wires also allows handset manufacturers to lower development costs by simplifying clamshell or sliding handset designs.

[0005] While MDDI and other data interfaces can be used to efficiently provide high speed data rates across interfaces, interface systems that exchange data received over an MDDI or other data interface link are often slow and not optimized for a particular application, such as, for example, processing camera images and control data to be exchanged between the lower and upper clamshell portions of a cellular telephone.

[0006] What is needed is a digital data device interface to provide efficient processing of data gathered and exchanged over an MDDI or other high speed link. Commonly owned, copending U.S. Patent Application No. _____ (having attorney docket number 1549.2350001), entitled *Digital Data Interface Device*, filed November 23, 2005, describes such a device. The present application describes a message format that can be used within a digital data interface device.

SUMMARY OF THE INVENTION

[0007] The present invention provides a digital data interface device message format that describes command and response messages to be exchanged between a digital device having a system controller and a digital data interface device. The digital data interface device includes a message interpreter, content module and a control module. The message interpreter module receives and interprets commands from and generates response messages through the communication link to a system controller, interprets the messages, and routes the information content of the commands to an appropriate module within the digital data interface device. The content

module receives data from a peripheral device, stores the data and transfers the data to the system controller through the communication link. The control module receives information from the message interpreter, and routes information to a control block of the peripheral device.

[0008] In one example, the digital data interface device includes an MDDI link controller. The digital data interface device can be used to control a peripheral device, such as a camera, bar code reader, image scanner, audio device or other sensor. In one particular example, a cellular telephone having a camera with an MDDI link and a digital data device interface is provided.

[0009] The digital data interface device message format includes a transaction identification field, a count field, a command identification field and a status field. Optionally, the message format can include a data field. When an MDDI link is used, a digital data interface device message can be included in an MDDI register access packet.

[00010] Further embodiments, features, and advantages of the invention, as well as the structure and operation of the various embodiments of the invention are described in detail below with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE FIGURES

[00011] The invention is described with reference to the accompanying drawings. In the drawings, like reference numbers indicate identical or functionally similar elements. The drawing in which an element first appears is indicated by the left-most digit in the corresponding reference number.

[00012] FIG. 1 is a diagram of a digital data device interface coupled to a digital device and a peripheral device.

[00013] FIG. 2 is a diagram of a message interpreter module.

[00014] FIG. 3 is a diagram of a content module.

[00015] FIG. 4 is a diagram of a control module.

[00016] FIG. 5 is a diagram of a cellular telephone having upper and lower clamshell sections that uses an MDDI interface to provide high speed data communications between electronics located in the upper and lower clamshells.

[00017] FIG. 6 is a diagram of the upper clamshell of a cellular telephone having a camera that uses an MDDI interface.

[00018] FIG. 7 is a diagram of a digital data interface device message format.

[00019] FIG. 8 is a diagram of a register access packet that includes a reverse encapsulation message containing a digital data interface device message.

DETAILED DESCRIPTION OF THE INVENTION

[00020] This specification discloses one or more embodiments that incorporate the features of this invention. The disclosed embodiment(s) merely exemplify the invention. The scope of the invention is not limited to the disclosed embodiment(s). The invention is defined by the claims appended hereto.

[00021] The embodiment(s) described, and references in the specification to "one embodiment", "an embodiment", "an example embodiment", etc., indicate that the embodiment(s) described may include a particular feature, structure, or characteristic, but every embodiment may not necessarily include the particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with an embodiment, it is understood that it is within the knowledge of one skilled in the art to effect such feature, structure, or characteristic in connection with other embodiments whether or not explicitly described.

[00022] Embodiments of the invention may be implemented in hardware, firmware, software, or any combination thereof. Embodiments of the invention may also be implemented as instructions stored on a machine-readable medium, which may be read and executed by one or more processors. A machine-readable medium may include any mechanism for storing or

transmitting information in a form readable by a machine (e.g., a computing device). For example, a machine-readable medium may include read only memory (ROM); random access memory (RAM); magnetic disk storage media; optical storage media; flash memory devices; electrical, optical, acoustical or other forms of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.), and others. Further, firmware, software, routines, instructions may be described herein as performing certain actions. However, it should be appreciated that such descriptions are merely for convenience and that such actions in fact result from computing devices, processors, controllers, or other devices executing the firmware, software, routines, instructions, etc.

[00023] FIG. 1 is a diagram of a digital data device interface 100 coupled to a digital device 150 and a peripheral device 180. Digital device 150 can include, but is not limited to, a cellular telephone, a personal data assistant, a smart phone or a personal computer. In general digital device 150 can include digital devices that serve as a processing unit for digital instructions and the processing of digital presentation data. Digital device 150 includes a system controller 160 and a link controller 170.

[00024] Peripheral device 180 can include, but is not limited to, a camera, a bar code reader, an image scanner, an audio device, and a sensor. In general peripheral 180 can include audio, video or image capture and display devices in which digital presentation data is exchanged between a peripheral and a processing unit. Peripheral 180 includes control blocks 190. When peripheral 180 is a camera, for example, control blocks 190 can include, but are not limited to lens control, flash or white LED control and shutter control.

[00025] Digital presentation data can include digital data representing audio, image and multimedia data.

[00026] Digital data interface device 100 transfers digital presentation data at a high rate over a communication link 105. In one example, an MDDI communication link can be used which supports bi-directional data transfer with a maximum bandwidth of 3.2 Gbits per second. Other high rates of data

transfer that are higher or lower than this example rate can be supported depending on the communications link. Digital data interface device 100 includes a message interpreter module 110, a content module 120, a control module 130 and a link controller 140.

[00027] Link controller 140, which is located within digital data interface 100, and link controller 170, which is located within digital device 150 establish communication link 105. Link controller 140 and link controller 170 may be MDDI link controllers.

[00028] The Video Electronics Standards Association ("VESA") MDDI Standard describes the requirements of a high-speed digital packet interface that lets portable devices transport digital images from small portable devices to larger external displays. MDDI applies a miniature connector system and thin flexible cable ideal for linking portable computing, communications and entertainment devices to emerging products such as wearable micro displays. It also includes information on how to simplify connections between host processors and a display device, in order to reduce the cost and increase the reliability of these connections. Link controllers 140 and 170 establish communication path 105 based on the VESA MDDI Standard.

[00029] U.S. Patent No. 6,760,772, entitled *Generating and Implementing a Communication Protocol and Interface for High Data Rate Signal Transfer*, issued to Zou et al. on July 6, 2004 ('772 Patent") describes a data interface for transferring digital data between a host and a client over a communication path using packet structures linked together to form a communication protocol for presentation data. Embodiments of the invention taught in the '772 Patent are directed to an MDDI interface. The signal protocol is used by link controllers, such as link controllers 140 and 170, configured to generate, transmit, and receive packets forming the communications protocol, and to form digital data into one or more types of data packets, with at least one residing in the host device and being coupled to the client through a communications path, such as communications path 105. The interface provides a cost-effective, low power, bi-directional, high-speed data transfer

mechanism over a short-range "serial" type data link, which lends itself to implementation with miniature connectors and thin flexible cables. An embodiment of link controllers 140 and 170 establishes communication path 105 based on the teachings of the '772 Patent. The '772 Patent is herein incorporated by reference in its entirety.

[00030] Furthermore, the host includes one of several types of devices that can benefit from using the present invention. For example, a host could be a portable computer in the form of a handheld, laptop, or similar mobile computing device, such as is depicted in as digital device 150. It could also be a Personal Data Assistant (PDA), a paging device, or one of many wireless telephones or modems. Alternatively, a host device could be a portable entertainment or presentation device such as a portable DVD or CD player, or a game playing device.

[00031] The host can reside as a host device or control element in a variety of other widely used or planned commercial products for which a high speed communication link is desired with a client. For example, a host could be used to transfer data at high rates from a video recording device to a storage based client for improved response, or to a high resolution larger screen for presentations. An appliance such as a refrigerator that incorporates an onboard inventory or computing system and/or Bluetooth connections to other household devices, can have improved display capabilities when operating in an internet or Bluetooth connected mode, or have reduced wiring needs for in-the-door displays (a client) and keypads or scanners (client) while the electronic computer or control systems (host) reside elsewhere in the cabinet. In general, those skilled in the art will appreciate the wide variety of modern electronic devices and appliances that may benefit from the use of this interface, as well as the ability to retrofit older devices with higher data rate transport of information utilizing limited numbers of conductors available in either newly added or existing connectors or cables.

[00032] At the same time, a client could comprise a variety of devices useful for presenting information to an end user, or presenting information from a

user to the host. For example, a micro-display incorporated in goggles or glasses, a projection device built into a hat or helmet, a small screen or even holographic element built into a vehicle, such as in a window or windshield, or various speaker, headphone, or sound systems for presenting high quality sound or music. Other presentation devices include projectors or projection devices used to present information for meetings, or for movies and television images. Another example would be the use of touch pads or sensitive devices, voice recognition input devices, security scanners, and so forth that may be called upon to transfer a significant amount of information from a device or system user with little actual "input" other than touch or sound from the user. In addition, docking stations for computers and car kits or desk-top kits and holders for wireless telephones may act as interface devices to end users or to other devices and equipment, and employ either clients (output or input devices such as mice) or hosts to assist in the transfer of data, especially where high speed networks are involved.

[00033] However, those skilled in the art will readily recognize that the present invention is not limited to these devices, there being many other devices on the market, and proposed for use, that are intended to provide end users with high quality images and sound, either in terms of storage and transport or in terms of presentation at playback. The present invention is useful in increasing the data throughput between various elements or devices to accommodate the high data rates needed for realizing the desired user experience.

[00034] The inventive MDDI and communication signal protocol may be used to simplify the interconnect between a host processor, controller, or circuit component (for example), and a display within a device or device housing or structure (referred to as an internal mode) in order to reduce the cost or complexity and associated power and control requirements or constraints of these connections, and to improve reliability, not just for connection to or for external elements, devices, or equipment (referred to as an external mode).

[00035] Wireless communication devices each have or comprise apparatus such as, but not limited to, a wireless handset or telephone, a cellular telephone, a

data transceiver, or a paging or position determination receiver, and can be hand-held, or portable as in vehicle mounted (including cars, trucks, boats, trains, and planes), as desired. However, while wireless communication devices are generally viewed as being mobile, it is also understood that the teachings of the invention are applicable to "fixed" units in some configurations. In addition, the teachings of the invention are applicable to wireless devices such as one or more data modules or modems which may be used to transfer data and/or voice traffic, and may communicate with other devices using cables or other known wireless links or connections, for example, to transfer information, commands, or audio signals. In addition, commands might be used to cause modems or modules to work in a predetermined coordinated or associated manner to transfer information over multiple communication channels. Wireless communication devices are also sometimes referred to as user terminals, mobile stations, mobile units, subscriber units, mobile radios or radiotelephones, wireless units, or simply as 'users' and 'mobiles' in some communication systems, depending on preference.

[00036] In the context of wireless devices, the present invention can be used with wireless devices that uses a variety of industry standards, such as, but not limited to cellular Analog Advanced Mobile Phone System (AMPS), and the following digital cellular systems: Code Division Multiple Access (CDMA) spread spectrum systems; Time Division Multiple Access (TDMA) systems; and newer hybrid digital communication systems using both TDMA and CDMA technologies. A CDMA cellular system is described in the Telecommunications Industry Association/Electronic Industries Association (TIA/EIA) Standard IS-95. Combined AMPS & CDMA systems are described in TIA/EIA Standard IS-98. Other communications systems are described in the International Mobile Telecommunications System 2000/Universal Mobile Telecommunications System or IMT-2000/UM standards, covering what are commonly referred to as wideband CDMA (WCDMA), cdma2000 (such as cdma2000 1x-rxtt cdma2000 1x, 3x, or MC

standards, for example) or TD-SCDMA. Satellite based communication systems also utilize these or similar known standards.

[00037] In other embodiments, link controllers 140 and 170 can both be a USB link controller or they both can include a combination of controllers, such as for example, an MDDI link controller and another type of link controller, such as, for example, a USB link controller. Alternatively, link controllers 140 and 170 can include a combination of controllers, such as an MDDI link controller and a single link for exchanging acknowledgement messages between digital data interface device 100 and digital device 150. Link controllers 140 and 170 additionally can support other types of interfaces, such as an Ethernet or RS-232 serial port interface. Additional interfaces can be supported as will be known by individuals skilled in the relevant arts based on the teachings herein.

[00038] Within digital data interface device 100, message interpreter module 110 receives commands from and generates response messages through communication link 105 to system controller 160, interprets the command messages, and routes the information content of the commands to an appropriate module within digital data interface device 100. FIG. 2 shows details of the architecture and function of message interpreter module 110.

[00039] Specifically, referring to FIG. 2, message interpreter module 110 includes a message interpreter controller 210, a response buffer 220 and a command buffer 230.

[00040] Message interpreter controller 210 reads and interprets incoming messages, generates register access and generates response messages. Incoming messages, for example, include instructions from digital device 150 to control peripheral 180. Response messages may include acknowledgement messages back to digital device 150 that an instruction was executed or not. Response messages can also include requests to read data from peripheral 180 and unsolicited control commands to digital device 150.

[00041] Response buffer 220 is coupled to message interpreter controller 210 and buffers response messages. A response buffer controller 225 can be

coupled between response buffer 220 and link controller 140 to regulate the flow of outgoing response messages to link controller 140.

[00042] Command buffer 230 is also coupled to message interpreter controller 210 and buffers incoming command messages. A command buffer controller 235 can be coupled between command buffer 230 and link controller 140 that regulates the flow of incoming command messages received from link controller 140. Command buffer controller 235 also identifies a valid command message and detects a new transaction within the valid command message. Command buffer controller 235 includes an error detection mechanism that examines a predefined unique identifier associated with a command message to detect one or more missing parts within a particular command message or within a set of command messages. In an example implementation the predefined unique identifier includes a single bit at the start of a command message.

[00043] Referring back to FIG. 1, content module 120 receives data from peripheral device 180, stores the data and transfers the data to system controller 160 through communication link 105. FIG. 3 shows further details of the architecture and function of content module 120.

[00044] Referring to FIG. 3, content module 120 includes a content buffer 310, a read control module 320, a write and sync control module 330 and a register block 340. Content buffer 310 stores data that has been received from peripheral device 180.

[00045] Read control module 320 manages the transfer of data from content buffer 310 to link controller 140. For example, read control module 320 can receive a request for data from digital device 150 over link controller 140. Read control module 320 can provide messages to digital device 150 indicating the size of the data and whether data is ready to be sent. When data is available, data can then either be transferred directly from content buffer 310 or directly through read control module 320.

[00046] Write and sync control module 330 manages the flow of data from peripheral device 180 to content buffer 310. Write and sync control module

330 includes a means for selectively writing some or all of the data received from peripheral device 180 to content buffer 310. Write and sync control module 330 also includes a means for examining sync pulses contained within received data to determine one or more data boundaries for distinguishing content. Additionally, write and sync module 330 can include a means for inserting timing information into data.

[00047] Register block 340 stores operational parameters that affect the behavior of at least one of content buffer 310, read control module 320 and write and sync control module 330. Register block 340 can also be coupled to message interpreter module 110 for receiving operational parameters. For example, register block 340 can store video data masks that can be used for decimation of a video signal or frame when peripheral device 180 is a camera. Similarly, operational parameters can include instructions for sub-sampling within frames and lines of a video signal, as well as instructions used to determine edges of a video signal. Parameters can also include pixels per line and window height and width information that is then used to dictate the behavior of write and sync control module 330 and read control module 320.

[00048] Referring back to FIG. 1, control module 130 receives information from message interpreter 130, and routes information to control blocks 190 of peripheral device 180. Control module 130 can also receive information from control blocks 190 and routes the information to the message interpreter module 110. FIG. 4 shows further details of the architecture and function of control module 130.

[00049] Referring to FIG. 4, control module 130 includes a control register block 410 and a peripheral control block 420. Control register block 410 contains registers that provide the control instructions for peripheral control block 420. Control register block 410 is coupled between message interpreter module 110 and peripheral control block 420. Peripheral control block 420 gathers peripheral control information from control register block 410 and uses that information to control peripheral device 180. For example, when peripheral device 180 is a camera, peripheral control block 420 can include

control blocks for flash or white LED control, shutter and exposure control, lens control and master control of the camera.

[00050] FIG. 5 is a block diagram of a cellular telephone 500 having upper and lower clamshell sections that uses an MDDI interface to provide high speed data communications between components located in the upper and lower clamshells. The following discussion related to cellular telephone 500 provides an illustrative example that further shows the utility of digital data interface device 100 and provides additional details related to its implementation and use. Based on the discussions herein, use of a digital data interface device 100 with other devices, for example, a personal digital assistant and other types of mobile phones, will be apparent and are within the spirit and scope of the invention.

[00051] Referring to FIG. 5, a lower clamshell section 502 of cellular telephone 500 includes a Mobile Station Modem (MSM) baseband chip 504. MSM 104 is a digital baseband controller. The invention is not limited to use with MSM baseband chip 504. In other embodiments, MSM baseband chip 504 could be another type of baseband processor, programmable digital signal processors (DSPs), or controllers. An upper clamshell section 514 of cellular telephone 500 includes a Liquid Crystal Display (LCD) module 516 and a camera module 518. Both lower clamshell section 502 and upper clamshell section 514 are encased in plastic as is typically used with cellular phones. Hinges 550 and 552 mechanically connect lower clamshell 502 to upper clamshell 514. Flexible coupling 554 provides electrical coupling between lower clamshell 502 and upper clamshell 514.

[00052] MDDI link 510 connects camera module 518 to MSM 504. A MDDI link controller can be provided for each of camera module 518 and MSM 504. Within cellular telephone 500, for example, an MDDI Host 522 is integrated into interface system 530 which is coupled to camera module 518, while an MDDI Client 506 resides on the MSM side of the MDDI link 510. In an embodiment, the MDDI host is the master controller of the MDDI link.

[00053] In cellular telephone 500 pixel data from camera module 518 are received and formatted into MDDI packets by interface system 530 using MDDI Host 522 before being transmitted onto MDDI link 510. MDDI client 506 receives the MDDI packets and re-converts them into pixel data of the same format as generated by camera module 518. The pixel data are then sent to an appropriate block in MSM 504 for processing.

[00054] Similarly, MDDI link 512 connects LCD module 516 to MSM 504. MDDI link 512 interconnects an MDDI Host 508, integrated into MSM 504, and an MDDI Client 520 integrated into interface system 532 which is coupled to LCD module 516. Display data generated by a graphics controller of MSM 504 are received and formatted into MDDI packets by MDDI Host 508 before being transmitted onto MDDI link 512. MDDI client 520 receives the MDDI packets and re-converts them into display data and processes the display data through interface system 532 for use by LCD module 516.

[00055] Interface systems 530 and 532 represent different embodiments of digital data device interface 100. In the case of interface system 530, digital data device interface 100 elements will be implemented to support data transfer of camera images and camera control functions for a camera. In the case of interface system 532, digital data device interface 100 elements will be implemented to support data display to an LCD and control functions for the LCD. Interface system 530 is further explained to illustrate an embodiment of digital data device interface 100 when used in a cellular telephone with a camera, such as cellular telephone 500 with camera module 518.

[00056] The relationship between the devices in FIG. 1 and cellular telephone 500 is as follows. Digital data device interface 100 is represented by interface system 530. Link controller 140 is represented by MDDI Host 522. Peripheral 180 is represented by camera module 518. System controller 160 is represented by MSM 504 and link controller 170 is represented by MDDI client 506.

[00057] FIG. 6 is a diagram of upper clamshell 514 and provides further details related to interface system 530 to highlight the example embodiment of digital

data device interface 100 as used within a cellular telephone with a camera. Interface system 530 includes MDDI host 522, camera message interpreter 602, camera video interface 604, I2C master 606, motor control 608 and flash/white LED timer 610. The I2C bus is a commonly used control bus that provides a communication link between circuits. The I2C bus was developed by Philips Electronics N.V. in the 1980s.

[00058] Recall that interface system 530 corresponds to digital data device interface 100. The components of interface system 530 correspond to the components of digital data device interface 100 in the following manner. Camera message interpreter 602 corresponds to message interpreter module 100. Camera video interface 604 corresponds to content module 120. Collectively, I2C master 606, motor control 608 and flash/white LED timer 610 correspond to control module 130.

[00059] Camera message interpreter 602 receives commands and generates response messages through MDDI host 522 to MSM 504. Camera message interpreter 602 interprets the messages and routes the information content to the appropriate block within interface system 530, which can be referred to as an MDDI camera interface device. Camera video interface 604 receives image data from camera 620, stores the image data, and transfers the image data to MDDI host 522. Collectively, I2C master 606, motor control 608 and flash/white LED timer 610 form a camera control block. In this case I2C master 606 provide controls for managing camera 620, motor control 608 provides controls for managing lens 622 (*e.g.*, lens zoom functions), and flash/white LED timer 610 provides controls for managing flash/white LED 624 (*e.g.*, flash brightness and duration.)

[00060] FIG. 7 shows a digital data interface device message format 700. Digital data interface device message format 700 can be used, for example, to format messages that exchange information and commands between digital data interface device 100 and digital device 150. Message format 700 includes a transaction identifier field 710, a count field 720, a command identification field 730, a status field 740, and a data field 750. In one example, transaction

identifier field 710, count field 720, command identification field 730, and status field 740 are each one byte. Data field 750 is an optional field that may or may not be present. When present data field 750 is either four or eight bytes. In other examples the field sizes can be other lengths, depending on specific messaging needs. The field size uses an 8-bit format for each byte. In other examples, the bit format can include other formats, such as, for example, a 4-bit or 16-bit format.

[00061] Using the above message formats, command and response messages can be formatted. There are two types of command messages: a write command and a read command. A write command is a message to execute a command, and a read command is a message to read information from one or more registers. There are three types of response messages: a write acknowledgment, a read response and an unsolicited message. A write acknowledgment is a response message indicating a successful register access. A read response message contains information that was read from one or more registers. In some instances a read response message can contain status indicators or signals that were not stored in a register. An unsolicited message is generated by, for example, digital data interface device 100 without a request by system controller 160.

[00062] When communications link 105 is an MDDI link, digital data device interface messages can be encapsulated within MDDI register access packets. Register access packets are defined within the VESA MDDI standard.

[00063] FIG. 8 shows register access packet format 800. Register access packet format 800 includes a packet length field 810, a packet type field 820, a client ID field 830, a read/write flag field 840, a register address field 850, a parameter cyclic redundancy check ("CRC") field 860, a register data list field 870 and a register data CRC field 880. With the exception of register address field 850 and register data list field 870, each field is two bytes. Register address field 850 is four bytes and register data list field 870 is a multiple of four bytes. A digital data device interface message conforming to digital data device interface message format 700 can be encapsulated in register data list

field 870. The specific uses of the other fields are not germane to the present invention, and are described more fully in the VESA MDDI standard.

[00064] In general, digital data interface device 100 receives commands from system controller 160 through MDDI reverse encapsulation packets. The command IDs are embedded in the packet and decoded by message interpreter module 110. The content of the commands is then sent to the appropriate block within digital data interface device 100. Similarly, message interpreter module 110 is also responsible for constructing the response messages to the system controller 160. These messages are either a response to a specific command of system controller 160, or an unsolicited message generated by digital data interface device 100 or peripheral device 180.

[00065] The use of an MDDI message to encapsulate a digital data device interface message is intended to provide an example of how digital data device interface messages can be encapsulated in other existing message types, and is not intended to limit the invention. Based on the teachings herein, individuals skilled in the relevant arts will be able to determine how to encapsulate digital data interface device messages into other types of messages.

[00066] Referring back to FIG. 7, the transaction ID field 710 is used to identify messages. The transaction ID field 710 can also be used to associate a command message with a corresponding response message. Additionally, transaction ID field 710 can include a byte containing a unique value that specifies an unsolicited response message. Referring to FIG. 1, system controller 160 assigns a transaction identifier that is used to populate this field and uses the transaction ID field 710 to recognize the response to a specific command.

[00067] The count field 720 is used to determine the length of a message. Count field 720 can also be used to determine the number of status and data field bytes in a message.

[00068] The command identification field 730 identifies the type of command to be executed. Each specific command ID is the value of the register base address for the specific section of digital data interface device 100. When

peripheral device 180 is a camera, such as depicted in FIG. 6, an example set of command IDs is as follows:

Command ID	Description
0x00	Digital device configuration command.
0x40	Camera interface control command
0x60	Lens control command
0x80	I2C command
0x90	Shutter control command
0xA0	Flash control command
0xB0	Three wire serial interface control command
0xC0	Phase Lock Loop (PLL) control command
0xD0-0xFF	Reserved commands

[00069] The status field 740 is used to determine whether to read from or write to a register block. Status field 740 can also be used to request an acknowledgment that indicates whether a command has been executed. Similarly, status field 740 can be used to specify whether a command has been executed successfully. In one example, bit 0 is used to identify whether the message is a read or write command. Bit 1 is used to indicate whether an acknowledgement is required. Bit 3 is used to provide an acknowledgment status.

[00070] When a message is a write command, data field 750 includes data to be written to one or more registers. In this case, the data is routed to a register block based on a value in command identification field 730. When a message is a response message, data field 750 includes data that was read from one or more registers. For an unsolicited response message, data field 750 includes data related to the event that caused the unsolicited response to be sent.

[00071] The following message encodings are provided as illustrative examples of possible encodings when peripheral device 180 is a camera, as depicted in FIG. 6.

[00072] The formats for command messages sent by system controller 160 to digital data device interface 100 for register control of digital data device interface 100 are as follows:

	Name	# of bits	Description

Byte 0	Transaction ID	8	Transaction ID assigned by system controller 160
Byte 1	Count	8	Total number of bytes in this message
Byte 2	Command ID	8	Digital data interface device 100 command ID. This value is different based on the blocks within digital data interface device 100 that is being addressed.
Byte 3	Status	8	Bit 0 – read/write: 0=write, 1=read Bit 1 – ack req: 0=no request, 1=request Bit 2 – ack status: 0=fail/error, 1=pass/success Bits 3-7 – reserved
Byte 4	Register value	8	Data content
Byte 5	Register value	8	Data content
...	Register value	8	Data content
Byte 11	Register value	8	Data content

[00073] A message using the above format can contain all the register setting/configuration information needed for digital data device interface 100. The command ID indicates the starting register address in digital data device interface 100 to be addressed. Digital data device interface 100 will automatically increment the register address for each consecutive byte until all register value bytes in the message are consumed. The count indicates the number of register accesses in the packet (including the read/write byte). The following table lists example register sets (and corresponding command ID value) in digital data device interface 100 that can be configured using messages conforming to the above format:

Command ID	Description
0x0000	Null
0x0100	Send link shutdown packet and wait for reset to clear condition after packet is sent.
0x0200	Power up: enable clocks
0x03XX	Hibernate: this command tells communications link 105 whether to transmit filler packets or to power down when there are no more packets to send. This will happen after a certain number of sub-frames containing only filler packets, sub-frame headers, periodic reverse encapsulation packets, or periodic roundtrip measurement packets. This number is passed in as the low order byte for this command. 0x##: number of empty sub-frames to wait before going into hibernate state. This can be a value between 1 and 255.

	The value of zero disables auto-hibernation.
0x0400	Resets link controller 140
0x050X	Display ignore/listen: This command tells the link controller 140 whether to listen to link controller 170. 0x00: listens for link controller 170 – default setting to allow for remote wake-up.
0x06XX	Send reverse encapsulation: this sends a reverse encapsulation packet that requests a display capabilities packet. The lower byte is passed through as the flags field of a reverse encapsulation packet. This allows for requests of client capabilities packets or status packets.
0x0700	Send roundtrip measurement packet: this command tells digital data interface device 100 to send a reverse timing packet at the next available time.

[00074] The following two tables provide illustrative examples showing how command messages can be formatted for supporting a peripheral device, such as a camera. The format for a I2C command message is as follows:

	Name	# of bits	Description
Byte 0	Transaction ID	8	Transaction ID assigned by system controller 160
Byte 1	Count	8	Total number of bytes in this message
Byte 2	Command ID	8	I2C command ID
Byte 3	Status	8	Bit 0 – read/write: 0=write, 1=read Bit 1 – ack req: 0=no request, 1=request Bit 2 – ack status: 0=fail/error, 1=pass/success Bits 3-7 – reserved
Byte 4	Camera module slave address	8	Least significant bit used as the read/write bit
Byte 5	Camera module register address	8	Address of the register in the camera module
Byte 6	Register value	8	Data content
...	Register value	8	Data content
Byte 11	Register value	8	Data content

[00075] The format for a flash control (e.g., using a white LED) command message is as follows:

	Name	# of bits	Description
Byte 0	Transaction ID	8	Transaction ID assigned by system controller 160

Byte 1	Count	8	Total number of bytes in this message
Byte 2	Command ID	8	Flash control command ID
Byte 3	Status	8	Bit 0 – read/write: 0=write, 1=read Bit 1 – ack req: 0=no request, 1=request Bit 2 – ack status: 0=fail/error, 1=pass/success Bits 3-7 – reserved
	White LED intensity	8	0x00: 20mA 0x01: 40mA 0x02: 60mA ... 0x18: 500mA
Byte 4	Red-eye reduction mode pulses	8	Number of red-eye reduction pulses prior to full discharge pulse. This parameter should be set to 0x01 for full discharge pulse
Byte 5	Pulse Duration	8	Duration of flash/strobe pulses in clock units.
Byte 6	White LED Duration	8	0x00: no change; the state of the LED does not change. 0x01: LED on for 1 frame time ... 0xFF: LED on for 256 frame times
Byte 7	Red-eye reduction pulse interval	8	Interval between red-eye reduction mode pulses in clock units.
Byte 8	White LED ON	8	0x00: white LED off 0x01: white LED on 0x02: flash/strobe full discharge 0x04: flash/strobe red-eye reduction
Bytes 9-11	Reserved	8	

[00076] As discussed above, commands received from system controller 160 can require acknowledgment or a return value response from digital data device interface 100. This is defined by setting the ACK required bit in byte 3 of a command message. The three types of response messages are an acknowledgment response message, which indicates successful access to a control register within digital data interface device 100, a read response message, which carries information read from a peripheral device and an unsolicited message, which is generated by digital data interface device 100 without a request from system controller 160.

[00077] The format for an acknowledgment response message is as follows:

	Name	# of bits	Description
Byte 0	Transaction ID	8	Transaction ID assigned by system controller 160
Byte 1	Count	8	Total number of bytes in this message

Byte 2	Command ID	8	Specific command ID (corresponding to initial message sent by system controller 160)
Byte 3	Status	8	Bit 0 – read/write: 0=write, 1=read Bit 1 – ack req: 0=no request, 1=request Bit 2 – ack status: 0=fail/error, 1=pass/success Bits 3-7 – reserved

[00078] The format for a read response message is as follows:

	Name	# of bits	Description
Byte 0	Transaction ID	8	Transaction ID assigned by system controller 160
Byte 1	Count	8	Total number of bytes in this message
Byte 2	Command ID	8	Specific command ID (corresponding to initial message sent by system controller 160)
Byte 3	Status	8	Bit 0 – (N/A) Bit 1 – (N/A) Bit 2 – ack status, (N/A) Bits 3-7 – reserved
Byte 4	Register value	8	Value read from digital data interface device 100 or peripheral registers.
Byte 5	Register value	8	Value read from digital data interface device 100 or peripheral registers.
...	Register value	8	Value read from digital data interface device 100 or peripheral registers.
Byte 11	Register value	8	Value read from digital data interface device 100 or peripheral registers.

[00079] The format for an unsolicited response message is as follows:

	Name	# of bits	Description
Byte 0	Transaction ID	8	Transaction ID assigned by system controller 160
Byte 1	Count	8	Total number of bytes in this message
Byte 2	Command ID	8	N/A – 0X00
Byte 3	Status	8	Bit 0 – (N/A) Bit 1 – (N/A) Bit 2 – ack status, (N/A) Bits 3-7 – reserved
Byte 4	Interrupt status	8	The status of interrupts in digital data interface device – design specific.
Byte 5	Link controller 140 status	8	Status of link controller 140 – design specific.
Byte 6	Shutter/flash execution complete	8	Command execution complete 0x01 – shutter execution command complete 0x02 –strobe execution command complete
Byte 7	Message content	8	Value read from digital data interface device 100 or peripheral registers.
...	Message content	8	Value read from digital data interface device

			100 or peripheral registers.
Byte 11	Message content	8	Value read from digital data interface device 100 or peripheral registers.

[00080] The above message formats were intended to provide illustrative examples, and not to limit the scope of the invention. Based on the teachings herein, individuals skilled in the relevant arts will be able to develop additional message formats depending on the specific application and peripherals being used.

Conclusion

[00081] Exemplary embodiments of the present invention have been presented. The invention is not limited to these examples. These examples are presented herein for purposes of illustration, and not limitation. Alternatives (including equivalents, extensions, variations, deviations, etc., of those described herein) will be apparent to persons skilled in the relevant art(s) based on the teachings contained herein. Such alternatives fall within the scope and spirit of the invention.

[00082] All publications, patents and patent applications mentioned in this specification are indicative of the level of skill of those skilled in the art to which this invention pertains, and are herein incorporated by reference to the same extent as if each individual publication, patent or patent application was specifically and individually indicated to be incorporated by reference.

WHAT IS CLAIMED IS:

1. A digital data interface device message format, comprising:
   a transaction identification field;
   a count field;
   a command identification field; and
   a status field.

2. A digital data interface device message format according to claim 1, wherein each of the fields comprises an 8 bit format.

3. A digital data interface device message format according to claim 1, further comprising a data field.

4. A digital data interface device message format according to claim 3, wherein the data field includes four or eight bytes.

5. A digital data interface device message format according to claim 3, wherein for a write command, the data field includes data to be written to one or more registers; and wherein for response messages, the data field includes data that was read from one or more registers.

6. A digital data interface device message format according to claim 3, wherein for an unsolicited response message, the data field includes data related to the event that caused the unsolicited response to be sent.

7. A digital data interface device message format according to claim 3, wherein the data field contains data that is routed to a register block based on a value of the command ID field.

8. A digital data interface device message format according to claim 1, wherein the transaction ID field comprises a byte that is used to identify a message.

9. A digital data interface device message format according to claim 8, wherein the transaction ID field comprises a byte that associates a command message with a corresponding response message.

10. A digital data interface device message format according to claim 8, wherein the transaction ID field comprises a byte containing a value that specifies an unsolicited response message; and wherein the byte value is unique.

11. A digital data interface device message format according to claim 3, wherein the count field comprises a byte that is used to determine the length of a message.

12. A digital data interface device message format according to claim 3, wherein the count field comprises a byte that is used to determine the number of status and data field bytes in a message.

13. A digital data interface device message format according to claim 1, wherein the status field comprises a byte used to determine whether to read from or write to a register block.

14. A digital data interface device message format according to claim 1, wherein the status field comprises a byte used to request an acknowledgment whether a command has been executed.

15. A digital data interface device message format according to claim 1, wherein the status field comprises a byte used to specify whether a command has been executed successfully.

16. A digital data interface device for transferring digital presentation data at a high rate between a host device and a client device over a communication link, comprising:

a message interpreter module that receives commands from and generates response messages through the communication link to a system controller, interprets the messages, and routes the information content of the commands to an appropriate module within the digital data interface device;

a content module that receives data from a peripheral device, stores the data and transfers the data to the system controller through the communication link;

a control module that receives information from the message interpreter, and routes information to a control block of the peripheral device; and

a set of messages, wherein each message comprises a transaction ID field, a count field, a command ID field, a status field and a data field.

17. The digital data interface device according to claim 16, wherein the set of messages includes messages for controlling a peripheral device.

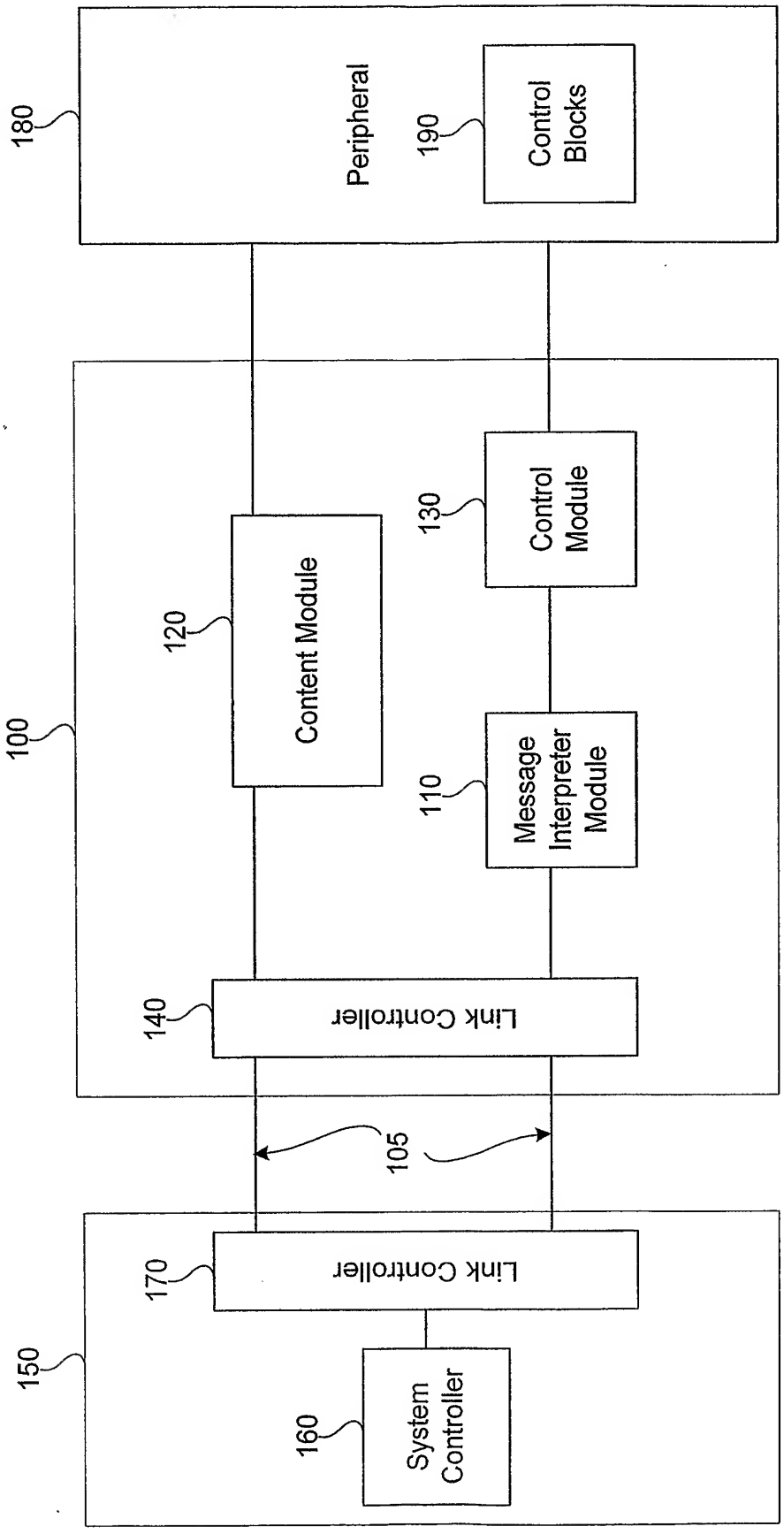

180
Peripheral
190
Control Blocks
100
120
Content Module
130
Control Module
110
Message Interpreter Module
140
Link Controller
105
150
170
Link Controller
160
System Controller

FIG. 1

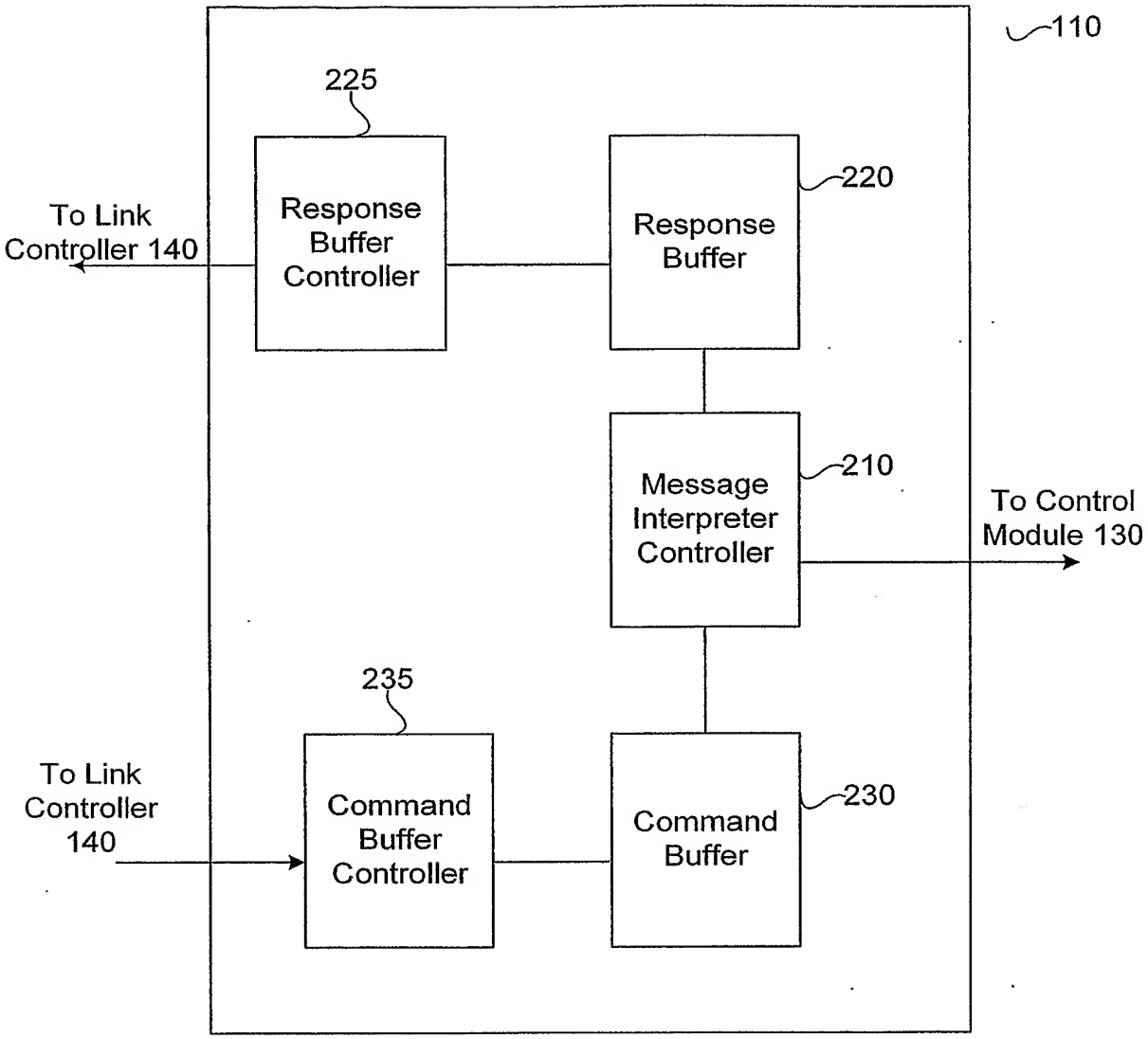

110
225
Response Buffer Controller
220
Response Buffer
To Link Controller 140
210
Message Interpreter Controller
To Control Module 130
235
Command Buffer Controller
230
Command Buffer
To Link Controller 140

FIG. 2

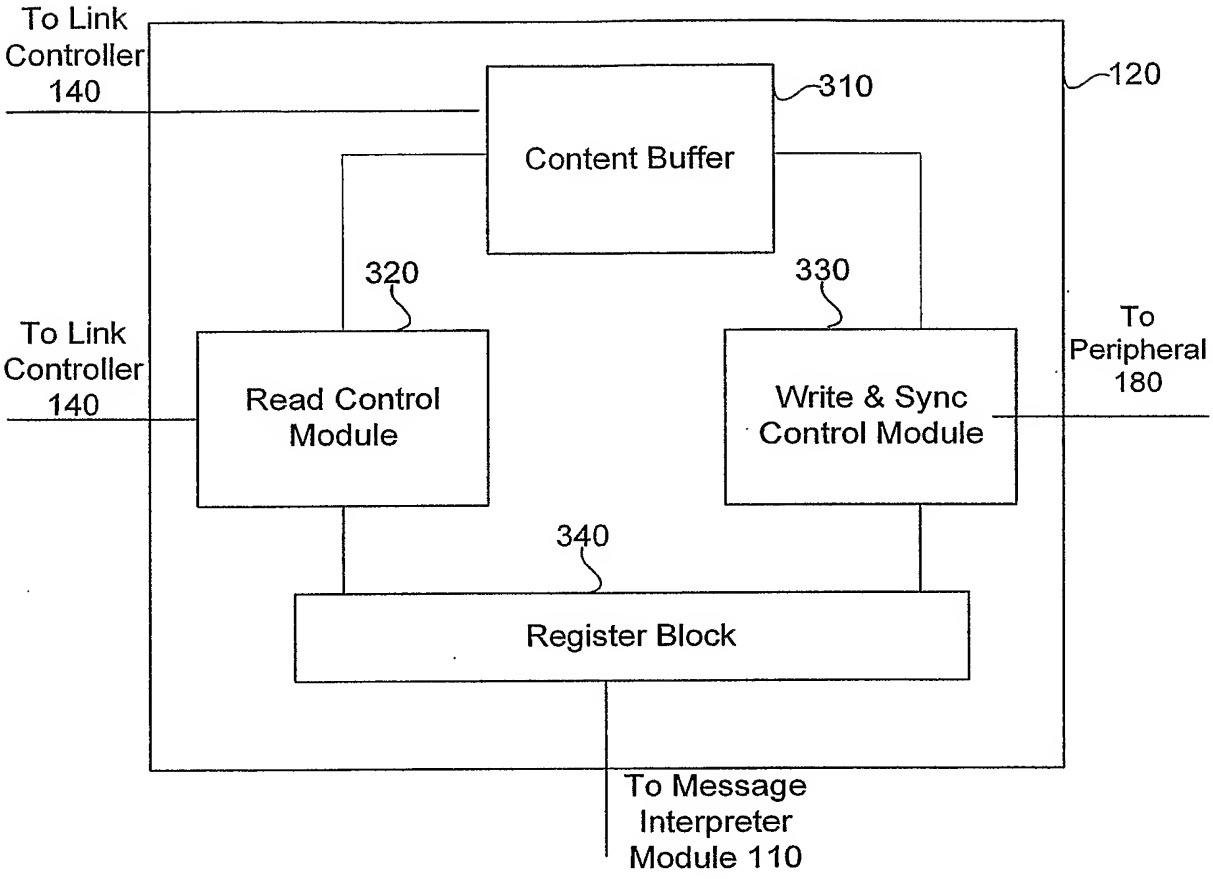

To Link Controller 140
Content Buffer
310
120
320
330
To Link Controller 140
Read Control Module
Write & Sync Control Module
To Peripheral 180
340
Register Block
To Message Interpreter Module 110

FIG. 3

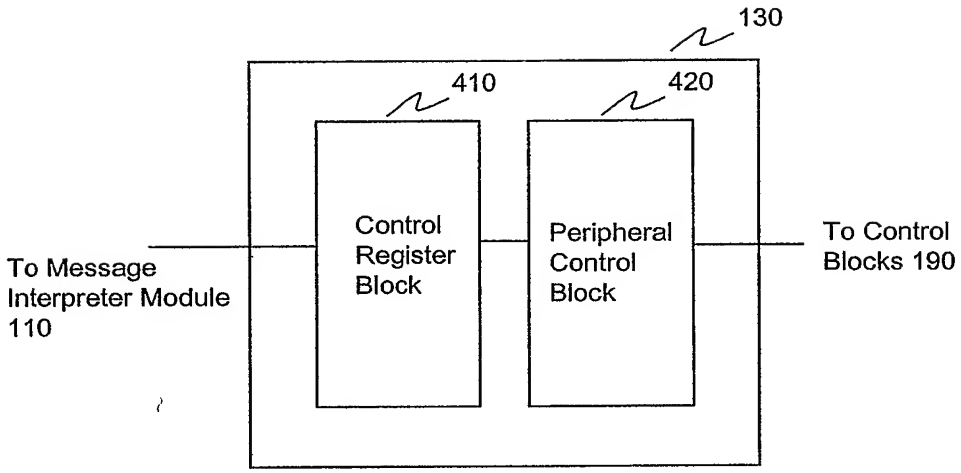

130
410
420
Control Register Block
Peripheral Control Block
To Message Interpreter Module 110
To Control Blocks 190

FIG. 4

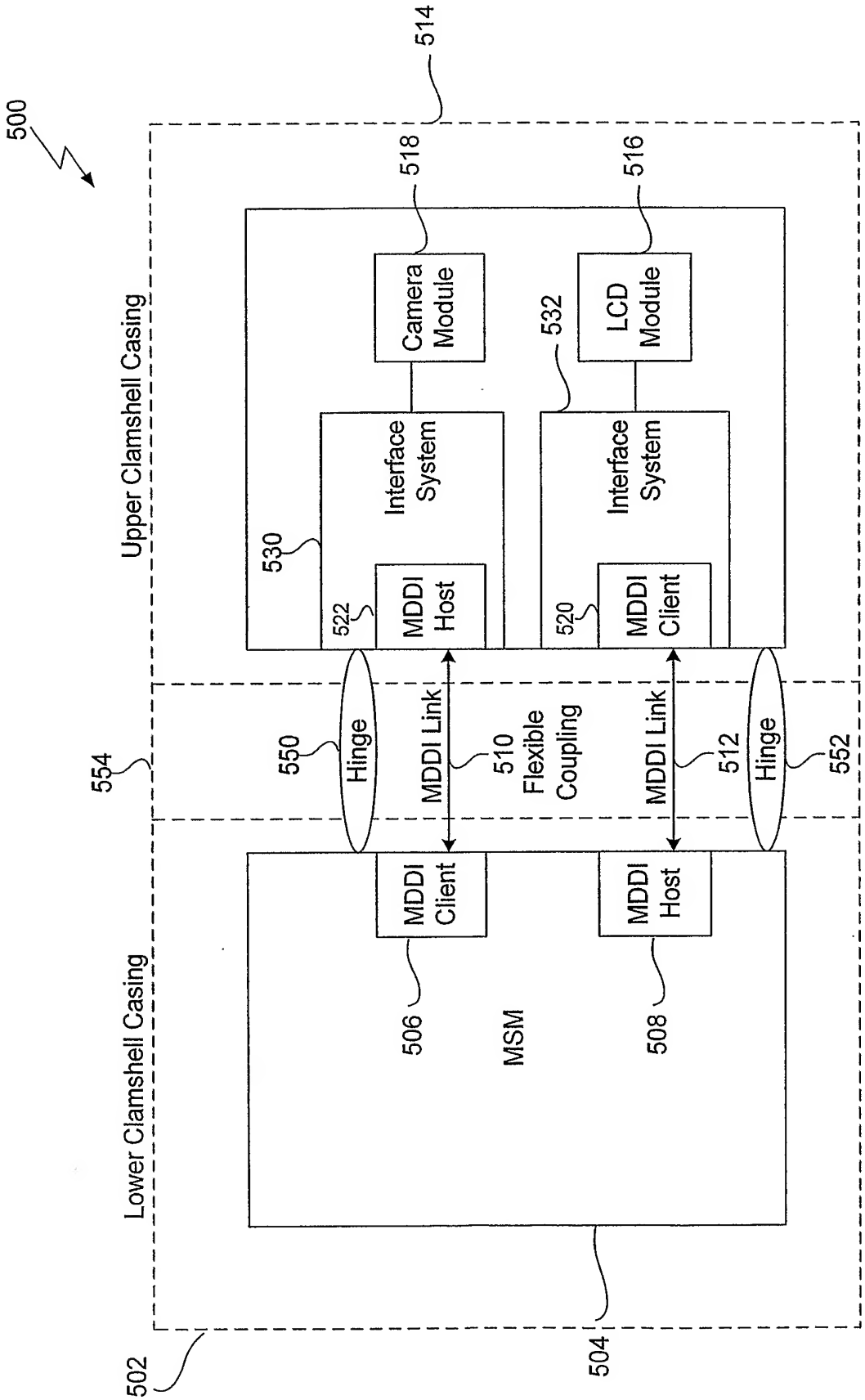

500
514
518
516
Camera Module
LCD Module
532
Interface System
Interface System
530
522
MDDI Host
520
MDDI Client
Upper Clamshell Casing
554
550
Hinge
MDDI Link
510
Flexible Coupling
MDDI Link
512
Hinge
552
MDDI Client
MDDI Host
506
MSM
508
Lower Clamshell Casing
502
504

FIG. 5

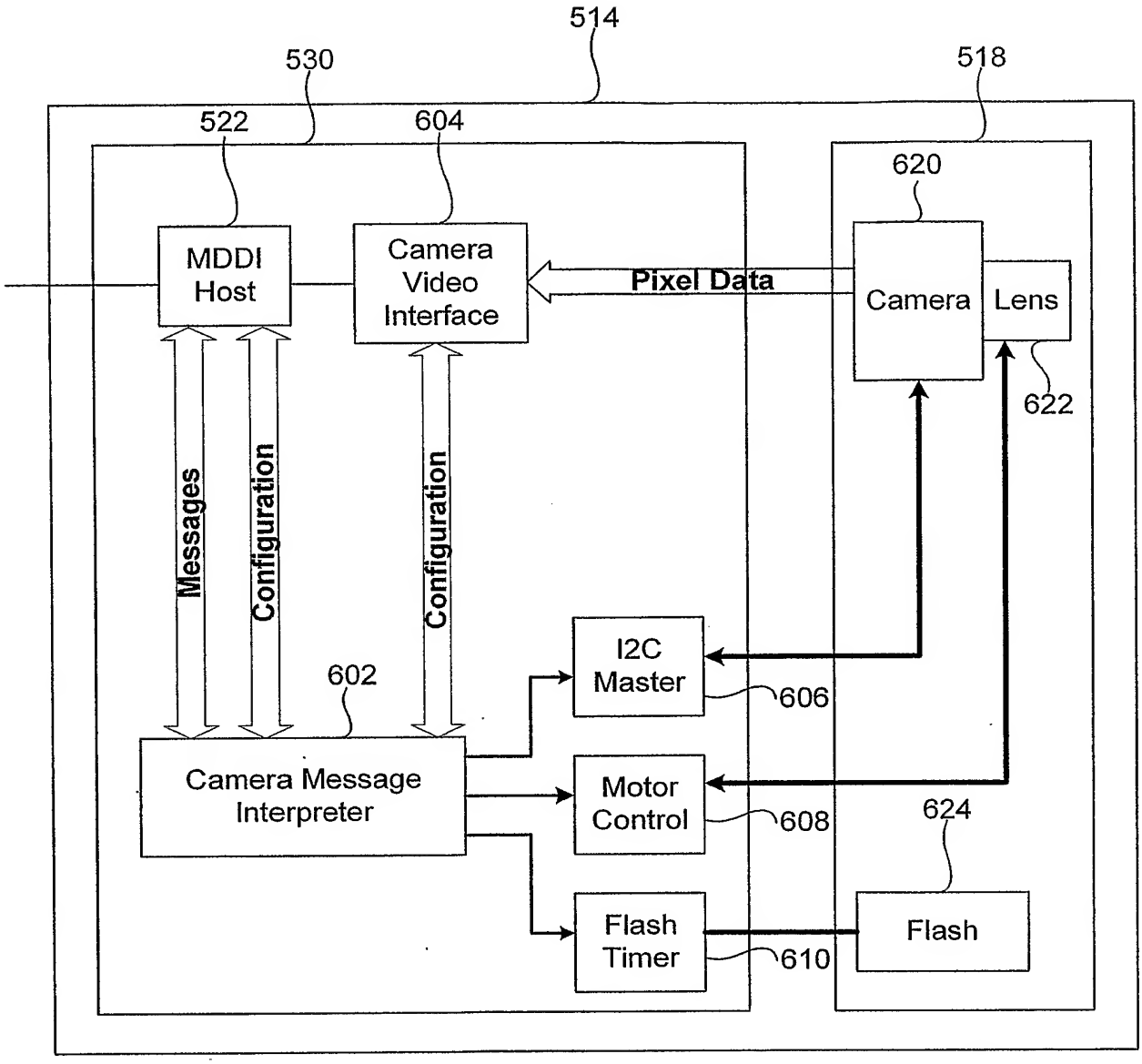

514
530
518
522
604
620
MDDI Host
Camera Video Interface
Pixel Data
Camera
Lens
622
Messages
Configuration
Configuration
602
I2C Master
606
Camera Message Interpreter
Motor Control
608
624
Flash Timer
610
Flash

FIG. 6

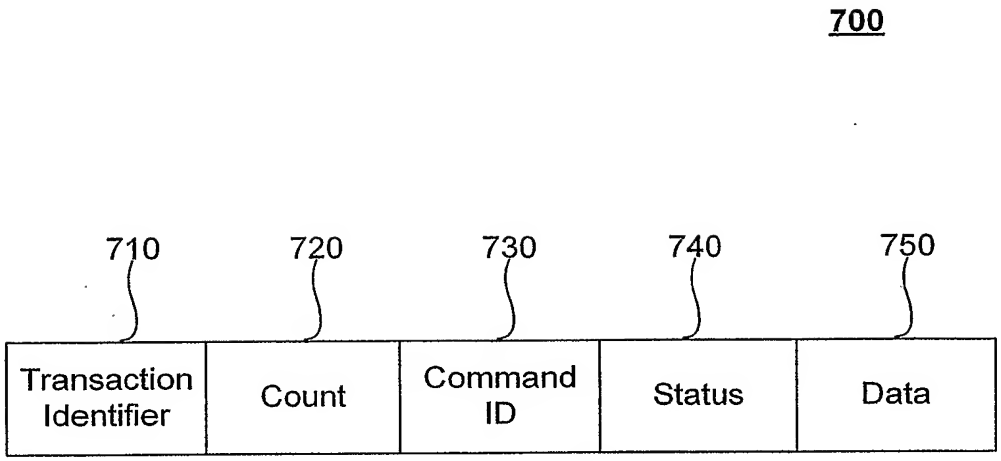

700
710
720
730
740
750
Transaction Identifier
Count
Command ID
Status
Data

FIG. 7

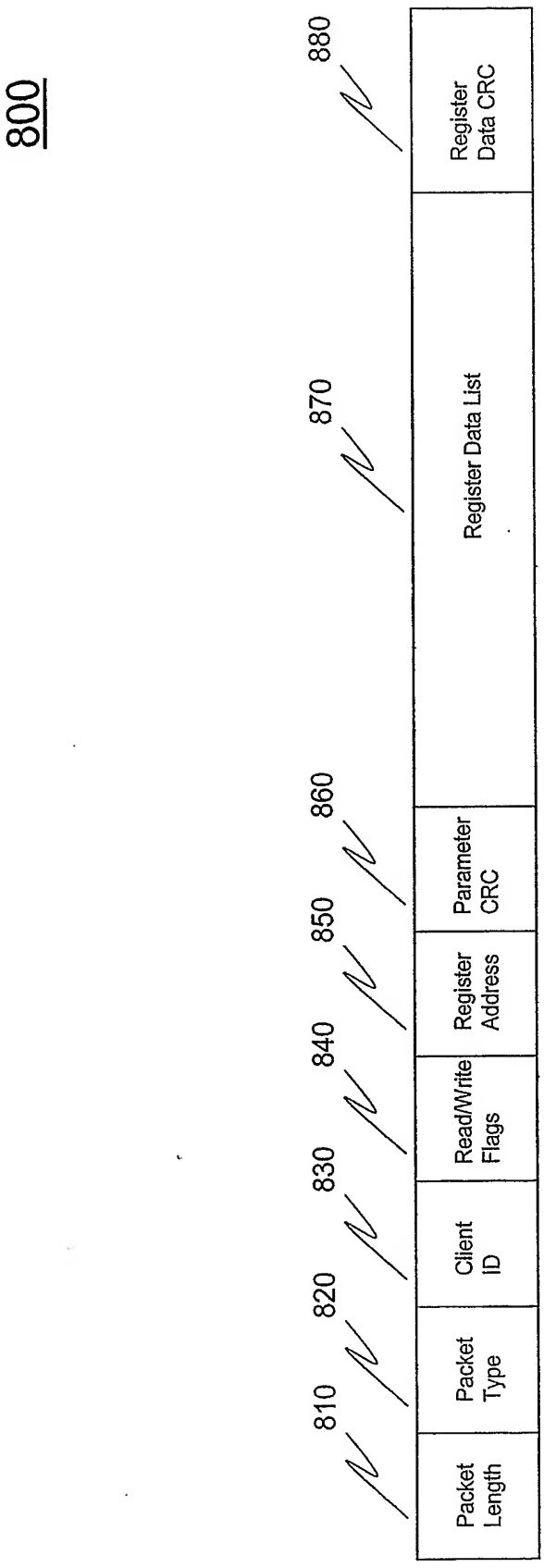

800
810
Packet Length
820
Packet Type
830
Client ID
840
Read/Write Flags
850
Register Address
860
Parameter CRC
870
Register Data List
880
Register Data CRC

FIG. 8